Sheet Generator

# Versão do Documento

Versão 1.0

# Linguagem de programação

* Python (versão ainda não foi decidida)

# Bibliotecas a utilizar

* pyaudio
* numpy
* matplotlib.pyplot
* turtle.py

# Algoritmos a utilizar – Must Have

## Funcionalidades Básicas

* Captar áudio pelo microfone
* Plotar das ondas sonoras em gráfico
  + Identificar melhor formato para exibição de ondas
* Implementar a taxa de atualização do gráfico de ondas sonoras ( Quanto maior , mais custo computacional )
  + Identificar melhor taxa para atualização das ondas
* Retornar valor em Hertz (Hz) de uma posição do gráfico
* Identificar picos do gráfico de ondas sonoras
* Retornar valor da frequência em Hertz (Hz) do pico identificado
* Identificar mudança de pico
* Retornar valor em milisegundos ou segundos da mudança de um pico para o outro

## Funcionalidades Intermediárias

* Normalizar ondas sonoras no que se diz respeito à intensidade do som
* Criar templates das páginas, onde serão inseridos os pentagramas, utilizando alguma biblioteca de plotagem
* Criar as figuras musicais através de bibliotecas de desenho para serem inseridas no template dos pentagramas
* Se necessário, integrar bibliotecas de plotagem e de desenho para gerar a partitura final
* Relacionar os conceitos musicais com as saídas das funcionalidades básicas
  + Retornar as conclusões e imediatamente desenhar o que foi retornado na partitura.
* Converter a partitura gerada para .pdf

## Funcionalidades Avançadas

* Gerar a partitura em tempo real
  + Otimizar ao máximo para que seja o mais instantâneo possível

# Algoritmos a utilizar – Nice to Have

## Funcionalidades Gerais

Não organizadas em ordem de prioridade

* Implementar animação de um “cursor” indicando o progresso da música
* Implementar animação para que a plotagem dos desenhos seja fluida e bonita
* Identificar várias melodias ao mesmo tempo ( mais de uma voz )
  + Identificar vários picos do gráfico de ondas sonoras
  + Retornar valor da frequência em Hertz (Hz) dos picos identificados
  + Identificar várias mudanças de picos
  + Retornar valor em milisegundos ou segundos das mudanças de vários picos
* Animação de load equanto o sistema gera o .pdf da partitura
* Possibilitar o usuário fazer correções na partitura gerada
* Gerenciar as informações da partitura e metadados
  + Nome do Arquivo
  + Título da partitura
    - Subtítulo
  + Autor
  + Direitos autorais
  + Descrição
* Possibilitar a inserção de letra na partitura
* Implementar conceitos de Machine Learning para que, a cada vez que o sistema for usado, melhore sua precisão ao gerar as partituras
  + Treinar o sistema com algumas situações mais complexas que ele pode se deparar
  + Treinar o sistema com situações em que esteja gerando a partitura de forma incorreta
  + Gerar uma porcentagem de 0% a 100% de confiança para cada nota e, ao final da execução, ressaltar os pontos mais críticos.
  + Guardar na memória tudo o que foi executado para que, depois que o usuário parar de executar, verificar se existe alguma melhoria que pode ser feita através do que o computador já aprendeu

# Tarefas importantes

* Categorizar timbres
* Definir um instrumento ideal para o desenvolvimento
* Organizar a exibição da interface gráfica em toda sua execução
  + Tela inicial
  + Tela de execução de uma seção
  + Tela de encerramento de uma seção